

**REMARKS**

Claims 1-18 are currently pending in the application. By this amendment, claims 1, 5, 6, 12, 14, 16 and 18 are amended and claims 19-20 are added for the Examiner's consideration. The foregoing separate sheets marked as "Listing of Claims" shows all the claims in the application, with an indication of the current status of each.

The Examiner has rejected claim 5 under 35 U.S.C. §112, second paragraph, because it refers to a "vertical pre-configuration" element which had been deleted from a parent claim. The above amendment overcomes this ground of rejection

The Examiner has rejected claims 1-4, 6-9 and 11-17 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 7,080,092 to Upton. This ground of rejection is respectfully traversed for the reasons that follow. Upton describes an "application view" system usable for integrating applications across enterprises. The problem addressed by Upton is the lack of effective standardization. Particular integration vendors have marketed their own proprietary solutions, but from the perspective of packaged application vendors there is no incentive to integrate their systems with these integration solutions (col. 2, lines 48-55), because each integration solution has its own application program interface (API) such that a packaged application vendor "cannot leverage development beyond a single integration server" (col. 2, lines 55-58). In short, the application vendor gets no mileage out of integrating the solution of one integration vendor, because he must start fresh and do the work of integrating the solution all over again with each of many different integration vendor solutions.

Upton overcomes this lack of leverage by the following technique. Upton notes the existence of a foundation for a "standardized approach to the development

of” appropriate “adapters”. That foundation is “the widespread acceptance of standards such as the Java 2 Platform, Enterprise Edition (J2EE)” (col. 3, lines 54-55), and in particular the J2EE Connector architecture (col. 3, lines 60-61), which provides “standardized approach for the development of adapters for all types of applications” (col. 3, lines 61-63). By placing adapters developed using these standards on a commonly available application server, each application can be integrated with the application server, “instead of requiring that each application be integrated with every other application” (col. 3, lines 51-53). An adapter can be used to invoke the functionality of a first application and expose that functionality (col. 3, lines 1-2). Upton then provides an “application view component” as a way for a second application to use the adapter to access the functionality of the first application exposed by the adapter (col. 3, lines 2-5). As will be evident to one skilled in the art, the Upton invention is enabled by the standardization of access to the adapter, so that the work of one adapter to expose the functionality of an application is accessible to all applications that comply with the standard (col. 3, lines 65-67). Each application can then develop its own interface to the various adapters needed for access to other applications, this interface being called an “application view” (col. 4, lines 9-11). Significantly, an application view does not require its users to have an intimate knowledge of the programming details defined in an adapter (col. 4, lines 16-24).

Thus the adapters of the Upton reference provide access to functionality provided by the various programs to be integrated. If the various applications are visualized as spokes on a wheel, the various adapters are hosted on an application server at the hub of the wheel. This is a much more efficient integration than for each application to separately and independently construct its own connection to the other applications in the wheel.

By contrast the present invention serves a different purpose. The services provided by the “server bullets” of the invention are not provided, as in Upton, by other applications on the spokes of a wheel. The functionality provided by a “server bullet” resides on the server and is available to an application on a client computer via an API. It should be noted – as explained in the present application in connection with Fig. 1D (page 4, line 17, to page 5, line 4) – that the present invention uses an application server, upon which these “server bullet” services are built, to make these services available for completion of an application (page 4, lines 21-22). It is an important aspect of the invention that these services be constructed to be easily scalable. Thus, they are single commands or narrowly focused tasks and are built directly on the server itself (page 4, lines 22-25). Thus, in the present invention the server is not a location for adapters which provide access to functionality available elsewhere (as in Upton), but rather is the location of pre-built services usable in constructing the application in the first instance, precisely to avoid the difficulties otherwise encountered in scaling an entire application (page 4, lines 24-25).

Whereas Upton takes existing applications and uses adapters on a server to enable integration of these applications, the present invention is concerned with an earlier stage involving development of the application in the first instance. Thus – under the paradigm of enterprise software development defined by the present invention – the “application” is “incomplete”, having been developed in the first instance with a view toward delaying the customization for a particular enterprise environment until the particular enterprise environment is known and a suitable collection of “server bullets” have been developed and tailored to that environment (page 4, lines 1-7). The Examiner argues that the applications in Upton are “incomplete” because they are not yet integrated into an enterprise environment. In Upton, the “integration” is provided by having a hub which serves to interconnect the various applications as spokes on a wheel.

But the present invention is not concerned with application to application integration. What the present invention provides is a platform and methodology for constructing enterprise environments with “server bullets” tailored to the particular enterprise environment to provide a core of services. This is a new paradigm for enterprise software development. The idea is that this kind of environment, across a variety of enterprises, will make it easy for application developers to develop applications which are “incomplete” in the sense that they are structured to avoid the conventional and prior art “customization” step (shown in Fig. 4A and described at page 9, lines 3-9). Instead, an enterprise (and, in general, a variety of enterprises, each potentially customers for an application developed using the methodology of the invention) develops “server bullet” services (i.e. “reusable service components”, item 430 in Fig. 4B; page 9, lines 12-13). The application is not “completed” until after this platform of “server bullet” services has been prepared for the enterprise environment (page 9, lines 9-15). The application is then “completed” by a step that links to the “server bullet” services provided by a particular enterprise environment. This methodology for application development has two main advantages. First, it avoids scalability issues by leaving to the enterprise environment itself – via “server bullets” – the task of providing the services that would need to be scaled. Of course, the precise services thus made available for a particular enterprise environment will not be known until after the enterprise environment has been prepared, which is why the application is left in an “incomplete” state (page 9, lines 20-21). Second, by virtue of this methodology, the “completion ... in lieu of customization” step (page 9, lines 13-15) is designed to come at the end and can be repeated easily, so that individual “incomplete” applications can not only be adapted to different enterprise environments but “re-adapted” to changes in a particular enterprise environment. See the discussion in connection with Fig. 4C (page 10, lines 1-11) for an explanation of

how “completion” in accordance with the invention differs from “customization” under the prior art.

It should be noted that the Upton invention could be used independently and at the same time as the present invention, without overlapping their distinctive and different inventive contributions. Upton is concerned with integration in the sense of making the functionality provided by a first application available to a second application, notwithstanding that the first and second applications may be legacy applications that don’t talk to each other. In the prior art of integrating applications into an enterprise environment, from Upton’s point of view, integration meant a labor intensive adaptation of each program to each other program. Each program had to be adapted multiple times; Upton avoids that repetition by having a common adapter, which is accessible to the other applications via a standardized interface.

By contrast, the present invention is not concerned with integration of particular applications, one with another, which have been brought together in an enterprise environment and must somehow be cobbled together, either by the prior “integration solution” art or by using Upton. Instead, the present invention provides a novel approach – in essence, a paradigm shift – to the design and construction of an enterprise environment having an appropriate complement of elemental “server bullets”, thereby enabling development of an entire class of “incomplete” applications that can be “completed” efficiently and scalably for each different enterprise environment.

This provides a two pronged marketplace dynamic attractive to both application developers and enterprise developers. Using the invention, application developers have a methodology for designing their applications to be easily “completed ... in lieu of customization” to a wide variety of enterprise environments, each “completion” being done “later” (page 9, line 14) after a suitable enterprise platform has been prepared with “server bullets. Similarly, developers of an

enterprise environment can use the invention to provide a highly scalable platform that is both tailored to serve the particular needs of the enterprise and also able to accommodate easy “completion” of “incomplete” applications.

While it is true that this approach of the present invention facilitates “integration” of the enterprise environment itself in terms of an overall design and development concept, this has nothing to do with the application-to-application integration addressed by Upton. Upton discloses nothing of the dynamic interaction between the development of enterprise environments prepared with reusable service components (“server bullets”) and the development of “incomplete” applications able to serve an ever wider range of such environments (page 8, line 17, to page 9, line 2) as libraries grow of reusable “server bullet” components and vertical pre-configurations of these components. If the invention is successful in the marketplace, “server bullet” components designed for a specified enterprise environment will become available in such libraries for adaptation and “reuse” in other enterprise environments in that enterprise area (page 4, lines 6-7). If marketplace economies permit (page 9, lines 22-26), these reusable libraries may also include vertical pre-configurations of these components, pre-configurations that are similarly adapted (page 4, lines 8-10), eventually replacing functionalities provided by holdover legacy applications (page 5, lines 5-9) and being bundled as full-featured applications, shrink-wrapped for marketing (page 5, lines 7-10).

Thus the “re-usability” of Upton’s adapters does not track the re-usability claimed for “server bullet” components. Upton itself notes that a single adapter can be accessed by a plurality of applications. In this sense an adapter is “re-used”, but this is the ordinary reuse characteristic of any computer software routine that is available to be called by a variety of other routines. And, of course, the “server bullets” of the present invention also have that aspect, although even in that aspect the present invention is distinguished because the services provided by the “server

bullets” are provided from the server itself, as stand-alone applications, whereas in Upton the actual services (made accessible by the adapter) are provided by another of the applications to be integrated. But the meaning of “reusable” that is properly and distinctly descriptive of the invention relates to the reuse of a “server bullet” component in other enterprise environments and in other vertical pre-configuration bundles. The claims have been amended to make this clear. Upton’s adapters have no such aspects to distinguish their limited and conventional re-usability.

In practical effect, the present invention treats the server as an “application server”, as a source of basic functionality, called “server bullets”. These “server bullets” are somewhat analogous to the basic functionalities provided to all applications by an operating system. In this sense the enterprise environment can be viewed as a higher level version of an operating system, where certain functionalities important to the enterprise can be more efficiently provided to applications by “server bullets”, instead of having these functionalities coded separately by and for each application. Further – if the novel paradigm implicit in the invention is adopted in the marketplace – the selection and configuration of these “server bullet” functionalities will be tailored over time to suit the particular demands of specified enterprise environments (page 9, lines 17-19). In the present invention, the “server bullet” functionalities are created by “building and deploying the most commonly used functions and services directly on the application server” (page 4, lines 22-24), thereby eliminating “the need to scale an entire application” (page 4, lines 24-25).

The success of the present invention depends, of course, on the acceptance of its advantages by both enterprise environments and applications serving those environments. The significance of changes in “business process requirements” under the methodology of the present invention is that these changes are relatively easily accommodated in revised code in the application residing on the client through reliance upon the use of “server bullets” that have been built by and for the enterprise

on the enterprise server. The use of these “server bullets” avoids “the need to scale an entire application” (page 4, lines 24-25). In the present invention, multiple applications may be efficiently scaled for revised business process requirements by recoding with reuseable “server bullets”.

The novel paradigm shift which is implied by the invention and upon which the marketplace success of the invention depends can be understood by reference to the long past development of computing technology. It will be recalled that much earlier in the history of computing, before operating systems existed, each application had to provide for itself all of the services required. Operating systems provided a significant advance, enabling applications to rely upon services provided by the operating system. The present invention provides a similar advance for enterprise environments, provided that the enterprises find it cost effective to take responsibility for “server bullets” (page 8, lines 24, to page 9, line 2) and application developers find it cost effective to structure their applications not only to rely upon these “server bullets” (page 8, lines 19-21) but also in such a way that the “completion” step of linking to these “server bullets” can be performed later, after the enterprise platform has been prepared (page 9, lines 13-15).

The Upton reference is not pertinent to the above described limitations of the invention. The independent claims have been amended to clarify these distinctive aspects of the invention.

In view of the foregoing, it is requested that the application be reconsidered, that claims 1-18 be allowed, and that the application be passed to issue.

Should the Examiner find the application to be other than in condition for allowance, the Examiner is requested to contact the undersigned at 703-787-9400 (fax: 703-787-7557; email: clyde@wcc-ip.com) to discuss any other changes deemed necessary in a telephonic or personal interview.

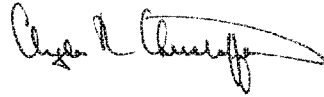


Server Bullet technology **AMENDMENT WITH RCE**  
Amendment dated 09/06/2007

07980001aa  
Reply to office action mailed 03/06/2007

If a further extension of time is required for this response to be considered as being timely filed, a conditional petition is hereby made for such extension of time. Please charge any deficiencies in fees and credit any overpayment of fees to Attorney's Deposit Account No. 50-2041.

Sincerely,

A handwritten signature in black ink, appearing to read "Clyde R. Christofferson", with a long, sweeping horizontal stroke at the end.

Clyde R Christofferson  
Reg. No. 34,138

Whitham, Curtis, Christofferson & Cook, P.C.  
11491 Sunset Hills Road, Suite 340  
Reston, VA 20190  
703-787-9400  
703-787-7557 (fax)

**Customer No. 30743**